

IN THE CLAIMS

Please amend claims 5 and 13 as follows:

1. (PREVIOUSLY PRESENTED) A computer-implemented method for displaying per-instance dynamic properties of an object comprising:
  - (a) receiving a reference to an object instance having a dynamic property that is created at runtime for the object instance on a per-instance basis and is not stored with the object;
  - (b) retrieving a reference to a property source instance from an association between the object and the property source instance, wherein the property source instance creates and supplies the dynamic property and an initial value for the dynamic property for/to the object instance; and
  - (c) providing the reference to the object instance and the reference to the property source instance to a control, wherein the control is configured to:
    - (i) retrieve the dynamic property from the property source instance; and
    - (ii) display the dynamic property in a user interface.
2. (ORIGINAL) The method of claim 1, wherein the dynamic property is provided by an application that is extending an object property set of the object.
3. (ORIGINAL) The method of claim 1, wherein the reference to the property source instance is retrieved from a mapping of property source instances to object class.
4. (ORIGINAL) The method of claim 1, wherein the control is further configured to: retrieve the standard properties for the object; and display the standard properties.
5. (CURRENTLY AMENDED) A computer-implemented system for displaying per-instance dynamic properties of an object comprising:
  - (a) a computer;
  - \_\_\_\_\_ (b) an application executing on the computer;
  - \_\_\_\_\_ (c) an object instance, in the application, having a dynamic property that is created at runtime for the object instance on a per-instance bases and is not stored with the object;

(~~b~~d) a property source instance, in the application, wherein the property source instance creates and supplies the dynamic property and an initial value for the dynamic property for/to the object instance;

(~~ee~~) an association, in the application, between the object and the property source instance; and

(~~ef~~) a host, in the application, configured to:

- (i) retrieve a reference to the object instance;
- (ii) retrieve a reference to the property source instance from the association; and
- (iii) provide the reference to the object instance and the reference to the property source instance to a control, wherein the control is configured to:

(1) retrieve the dynamic property from the property source instance;

and

(2) display the dynamic property in a user interface.

6. (ORIGINAL) The system of claim 5, wherein the dynamic property is provided by an application that is extending an object property set of the object.

7. (ORIGINAL) The system of claim 5, further comprising a mapping of property source instances to object classes, wherein the host is configured to retrieve the reference to the property source instance from the mapping.

8. (ORIGINAL) The system of claim 5, wherein the control is further configured to: retrieve the standard properties for the object; and display the standard properties.

9. (ORIGINAL) A computer-implemented method for providing a custom graphical user interface for editing a property of an object, comprising:

receiving a first object having a first property, wherein the first object provides a custom ActiveX control that defines a first user interface for displaying and editing the first property;

creating a list of one or more object properties to be displayed, wherein the list includes the first property;

instantiating the custom ActiveX control; and

displaying the object properties in the list, wherein the display of the first property comprises the first user interface defined by the instantiated custom ActiveX control, wherein the property may be edited through the first user interface.

10. (ORIGINAL) The method of claim 9, further comprising instantiating one or more stock ActiveX controls that define one or more additional user interfaces for displaying and editing remaining object properties in the list, wherein the stock ActiveX controls are not provided by any object containing one or more of the remaining object properties.

11. (ORIGINAL) The method of claim 10, wherein the first user interface and additional user interfaces are displayed in a single dialog box.

12. (ORIGINAL) The method of claim 9, wherein an application programming interface provides the ability to push the first object to a second object for display.

13. (CURRENTLY AMENDED) A system for providing a custom graphical user interface for editing a property of an object comprising:

(a) a computer;

(b) an application executing on the computer;

(c) one or more objects, in the application, wherein each object has one or more object properties;

(~~b~~d) a property inspector, in the application, configured to:

(i) interrogate the one or more objects to discover one or more object properties to be displayed;

(ii) create a list of the one or more object properties to be displayed; and

(iii) instantiate and host one or more property editors;

(~~ee~~) one or more property editors, in the application, wherein:

- (i) one of the property editors comprises a custom ActiveX control specified by one of the objects; and
- (ii) the custom ActiveX control defines a custom graphical user interface for displaying and editing one of the object properties.

14. (ORIGINAL) The system of claim 13, wherein one the property editors is comprised of a stock ActiveX control that defines an additional user interface for displaying and editing one or more additional properties in the list, wherein the stock ActiveX control is not provided by one of the objects that contains the one or more additional properties.

15. (ORIGINAL) The system of claim 14, wherein the custom graphical user interface and additional user interfaces are displayed in a single dialog box.

16. (ORIGINAL) The system of claim 13, further comprising an application programming interface configured to provide the ability to push the one or more object to the property inspector for display.

17. (PREVIOUSLY PRESENTED) A computer-implemented system for displaying per-instance dynamic properties of an object comprising:

- (a) an object instance of a class, wherein:
  - (i) an initial value for one or more static properties of the class are assigned at run time; and
  - (ii) the object instance has a dynamic property and an initial value of the dynamic property that are both generated and supplied, by a property source instance, at runtime for the object instance, on a per-instance basis and are not stored with the object;
- (b) an association between either:
  - (i) the object instance and the property source instance; or
  - (ii) the class and the property source instance; and

(c) a user interface component that displays a collection of properties of the object instance including the one or more static properties and the dynamic property on a display device, wherein the user interface component is configured to:

- (i) retrieve a reference to the object instance;
- (ii) retrieve the one or more static properties from the object instance;
- (iii) access the association to determine the property source instance associated with the object instance;
- (iv) call a method of the determined property source instance with the reference to the associated object instance;
- (v) receive the dynamic property, from the property source instance, wherein the property source instance dynamically generated the dynamic property and an initial value for the dynamic property; and
- (vi) display the static property and the dynamic property on the display device.

18. (PREVIOUSLY PRESENTED) The method of claim 9, wherein:  
custom ActiveX controls are provided on a per-property basis;  
two or more object properties are in the properties list; and  
the two or more object properties are displayed in a list, wherein each of the two or more object properties are displayed using user interfaces defined by the custom ActiveX controls.